**LENGUAJE DE DEFINICIÓN DE DATOS (DDL)**

Sentencias

1. **CREATE**: La sentencia CREATE TABLE se utiliza para crear una tabla dentro de la cual habrá columnas que contienen datos y restricciones

**CREATE TABLE** <nombre de tabla>

(nombre\_columna1 tipo [restricción de columna],

........

nombre\_columnaN tipo [restricción de columna],

[restricción\_de\_tabla]);

**Ejemplo:**

CREATE TABLE Empleados (

id NUMBER(6),

apellidos VARCHAR2(40),

nombre VARCHAR2(20) NOT NULL,

…

supervisor\_id NUMBER(6),

CONSTRAINT emp\_pk PRIMARY KEY (id)

);

**Claves foráneas en MySQL**

Estrictamente hablando, para que un campo sea una clave foránea, éste necesita ser definido como tal al momento de crear una tabla. Se pueden definir claves foráneas en cualquier tipo de tabla de MySQL, pero únicamente tienen sentido cuando se usan tablas del tipo InnoDB.

Para trabajar con claves foráneas, necesitamos hacer lo siguiente:

\* Crear ambas tablas del tipo InnoDB.

\* Usar la sintaxis FOREIGN KEY(campo\_fk) REFERENCES nombre\_tabla (nombre\_campo)

\* Crear un índice en el campo que ha sido declarado clave foránea.

InnoDB no crea de manera automática índices en las claves foráneas o en las claves referenciadas, así que debemos crearlos de manera explícita. Los índices son necesarios para que la verificación de las claves foráneas sea más rápida. A continuación se muestra como definir las dos tablas de ejemplo con una clave foránea.

CREATE TABLE cliente

(

id\_cliente INT NOT NULL,

nombre VARCHAR(30),

PRIMARY KEY (id\_cliente)

) TYPE = INNODB;

CREATE TABLE venta

(

id\_factura INT NOT NULL,

id\_cliente INT NOT NULL,

cantidad INT,

PRIMARY KEY(id\_factura),

INDEX (id\_cliente),

FOREIGN KEY (id\_cliente) REFERENCES cliente(id\_cliente)

) TYPE = INNODB;

1. Ver la estructura de una tabla

**DESCRIBE** <nombre de tabla>

1. Renombrar una tabla

**RENAME TABLE** <nombre tabla existente> TO <nuevo nombre tabla>

1. Eliminar una tabla de la base de datos

**DROP TABLE** <nombre tabla> [CASCADE, RESTRICT]

Ejemplos:

– DROP TABLE DEPARTAMENTOS CASCADE

(La tabla se borra, así como las posibles restricciones relativas a esta tabla)

– DROP TABLE DEPARTAMENTOS RESTRICT

(La tabla se borra sólo si no se hace referencia a ella en ninguna restricción, p.e. en la definición de claves ajenas)

1. Modificar una tabla

**ALTER TABLE <nombre tabla>**

La sentencia **Alter Table** se utiliza para cambiar una tabla existente. Dentro de la tabla podemos **Add** (añadir) o **Drop** (borrar) columnas y restricciones

(PRIMARY KEY,FORING KEY, UNIQUE, CHECK CONSTRAINT).

{ ADD <nombre columna nueva> <tipo de dato> [NOT NULL]

MODIFY <nombre columna> [DEFAULT | DROP DEFAULT] valor

DROP <nombre columna> [CASCADE | RESTRICT]

ADD [PRIMARY KEY (nombre columna) |FOREIGN KEY (nombre columna) REFERENCES nombre\_tabla | UNIQUE (nombre columna) | CHECK (condición)

DROP CONSTRAINT nombre-restricción [CASCADE| RESTRICT]

**EJEMPLOS**

**Agregar a la tabla areas el campo Responsable de tipo char(30)**

*alter table areas*

*ADD responsable char(30) not null;*

**Modificar el campo nombre de la tabla departamentos a char(50)**

*alter table departamentos*

*MODIFY nombre char(50);*